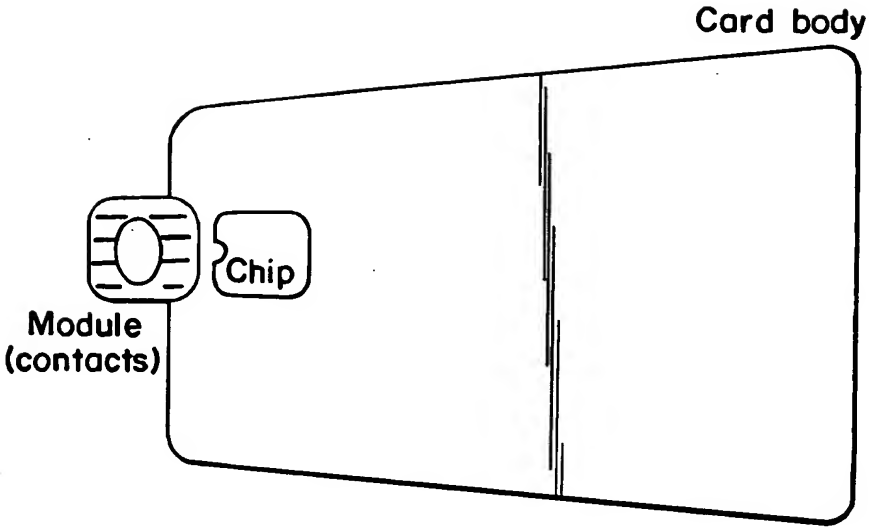

Card body
Chip
Module
(contacts)

Fig. IA

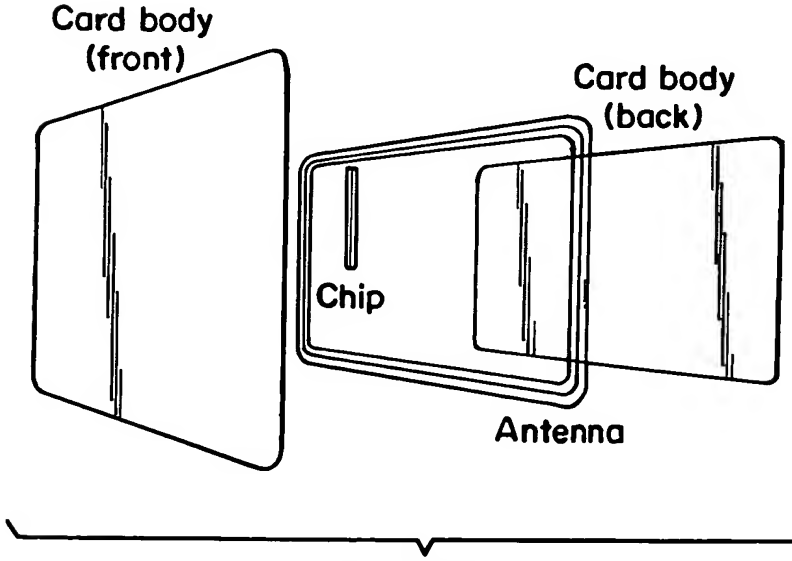

Card body
(front)
Card body
(back)
Chip
Antenna

Fig. IB

Fig. 1C

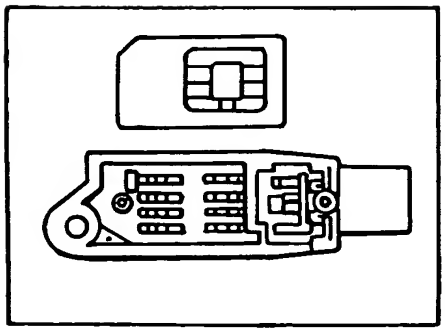


Fig. 1D

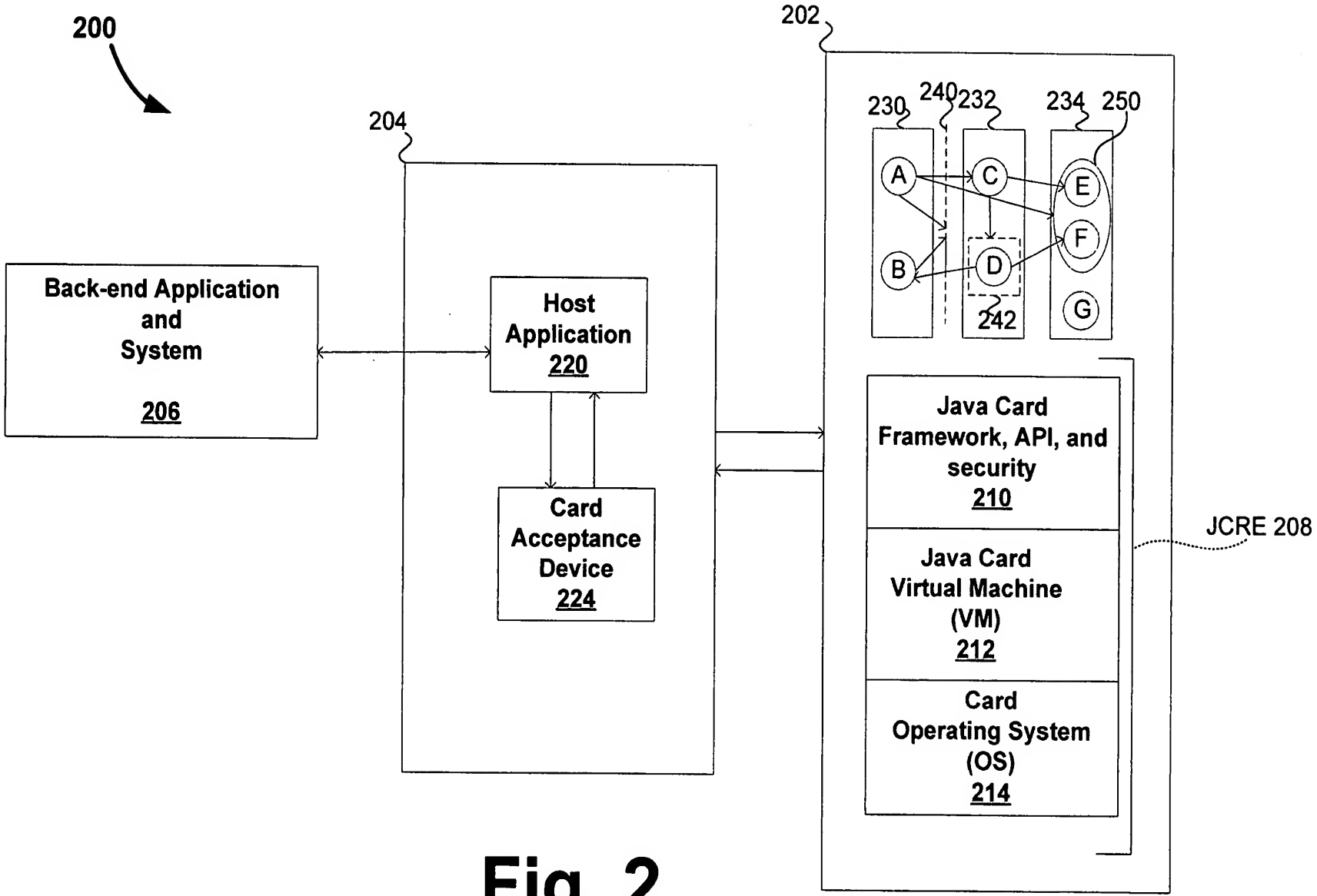

200
202
204
230
240
232
234
250
A
B
C
D
E
F
G
242
Back-end Application and System
206
Host Application
220
Card Acceptance Device
224
Java Card Framework, API, and security
210
Java Card Virtual Machine (VM)
212
Card Operating System (OS)
214
JCRE 208

Fig. 2

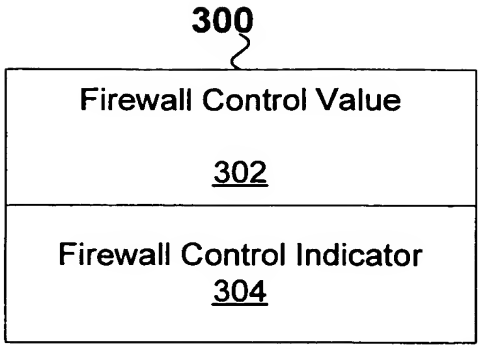


Fig. 3A

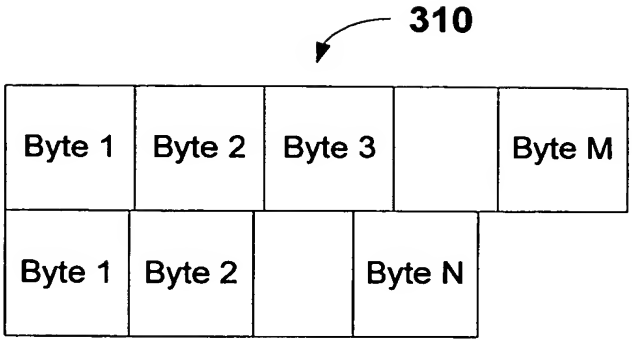


Fig. 3B

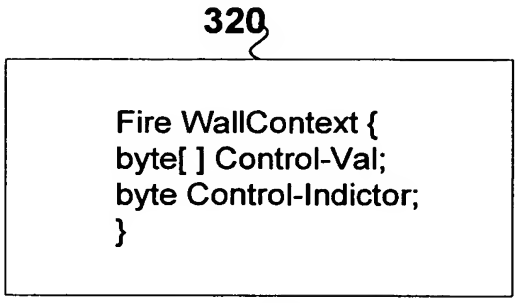


Fig. 3C

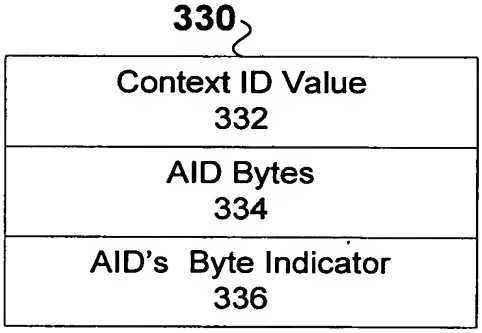


Fig. 3D

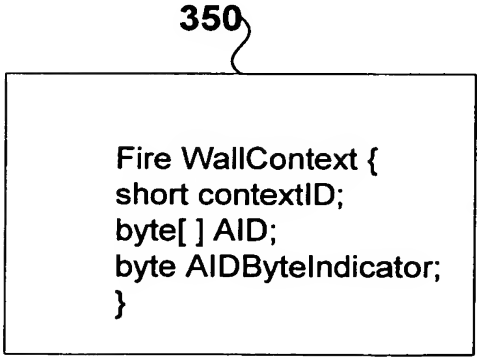


Fig. 3F

Fig. 3E

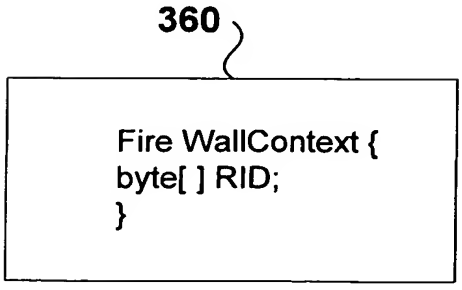


Fig. 3G

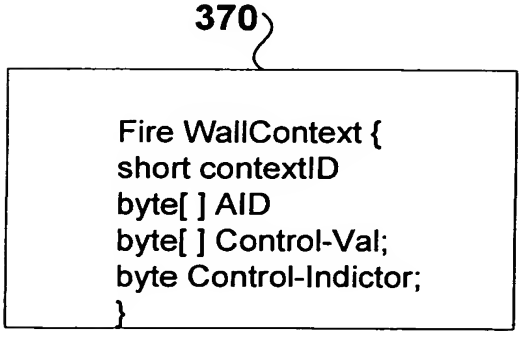
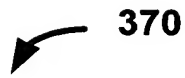


Fig. 3H



Firewall Control Value 1 372	Firewall Control Value 2 376
Firewall Control Indicator 1 374	Firewall Control Indicator 2 378

Fig. 3I

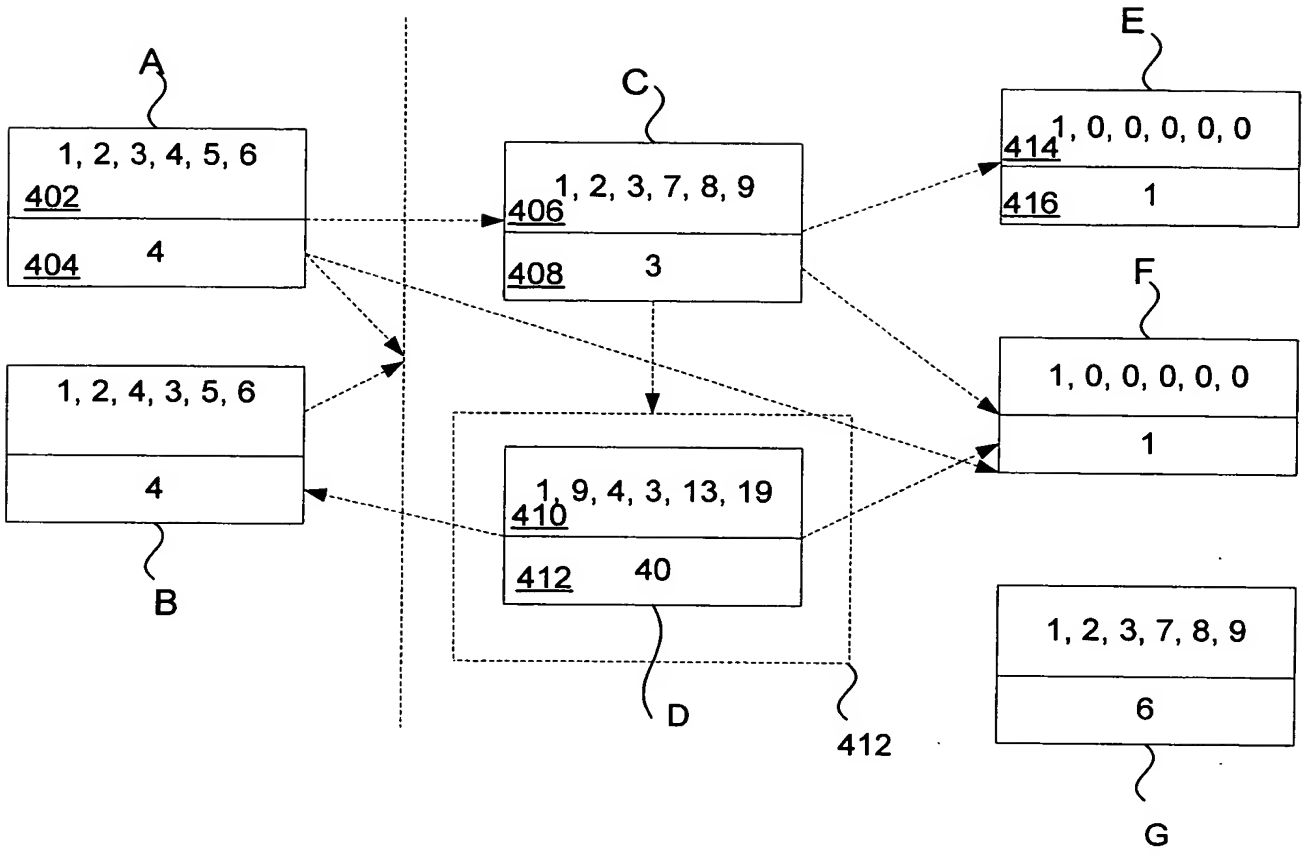

A
1, 2, 3, 4, 5, 6
402
404
4
B
1, 2, 4, 3, 5, 6
4
C
1, 2, 3, 7, 8, 9
406
408
3
D
1, 9, 4, 3, 13, 19
410
412
40
412
E
1, 0, 0, 0, 0, 0
414
416
1
F
1, 0, 0, 0, 0, 0
1
G
1, 2, 3, 7, 8, 9
6

Fig. 4

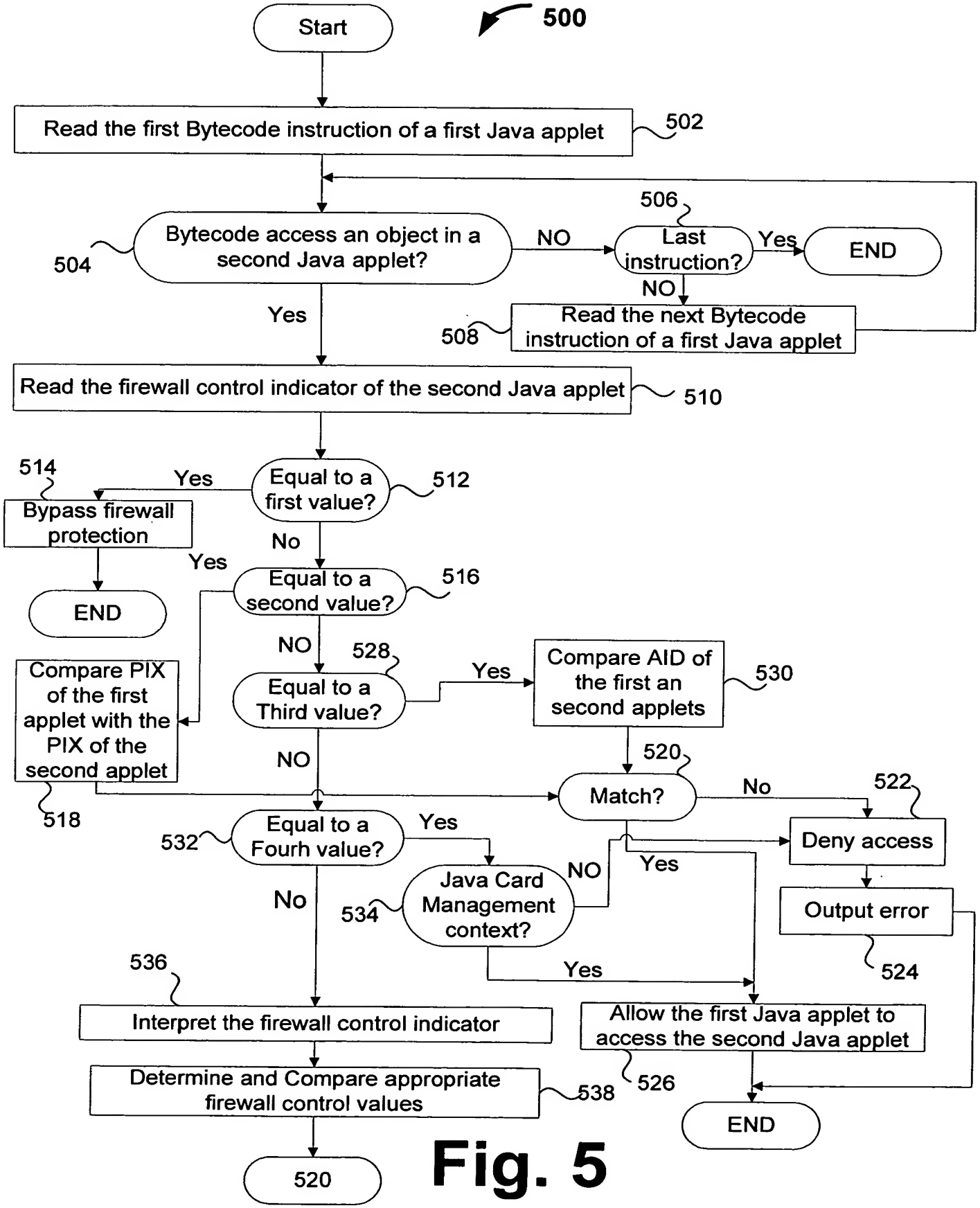

Start
500
Read the first Bytecode instruction of a first Java applet
502
Bytecode access an object in a second Java applet?
504
NO
506
Last instruction?
Yes
END
NO
Yes
508
Read the next Bytecode instruction of a first Java applet
Read the firewall control indicator of the second Java applet
510
514
Yes
Equal to a first value?
512
Bypass firewall protection
No
Yes
Equal to a second value?
516
END
NO
528
Compare PIX of the first applet with the PIX of the second applet
Equal to a Third value?
Yes
Compare AID of the first an second applets
530
518
NO
520
Match?
No
522
Equal to a Fourh value?
532
Yes
Deny access
NO
Yes
Java Card Management context?
534
Output error
No
524
Yes
536
Interpret the firewall control indicator
Allow the first Java applet to access the second Java applet
526
Determine and Compare appropriate firewall control values
538
END
520

Fig. 5

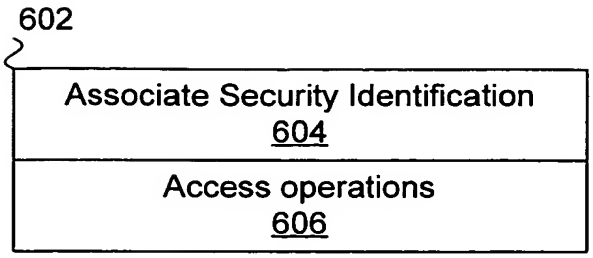


Fig. 6A

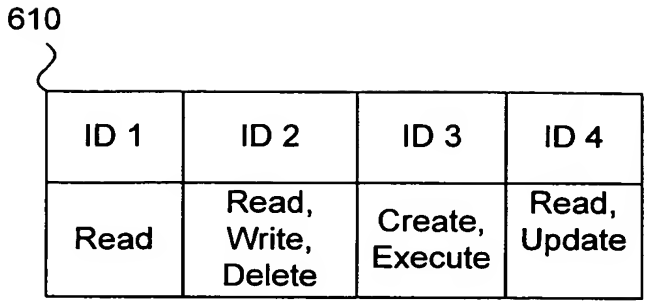


Fig. 6B

```
Fire WallContext {
Context ID;
Associated ID AssociatedIDAccess[ ];
                }


AssociatedID {
ValidID
AccessMask;
 }
```

Fig. 6C

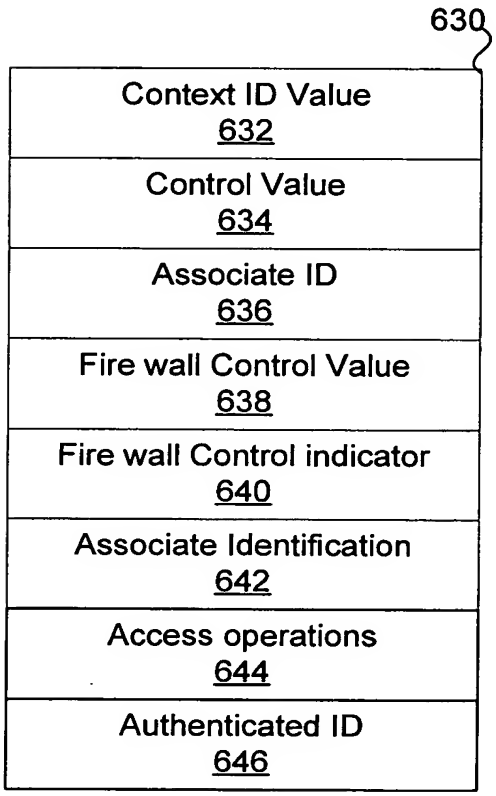


Fig. 6D

```
Fire WallContext {
Context ID;
boolean control-value-access;
boolean Assocaite-ID-access;
byte[ ] Control-Val;
byte Control-Indictor;
Associated ID AssociatedIDAccess[ ];
 }


AssociatedID {
ValidID
AccessMask;
boolean Authenticated-Already
 }
```

Fig. 6E

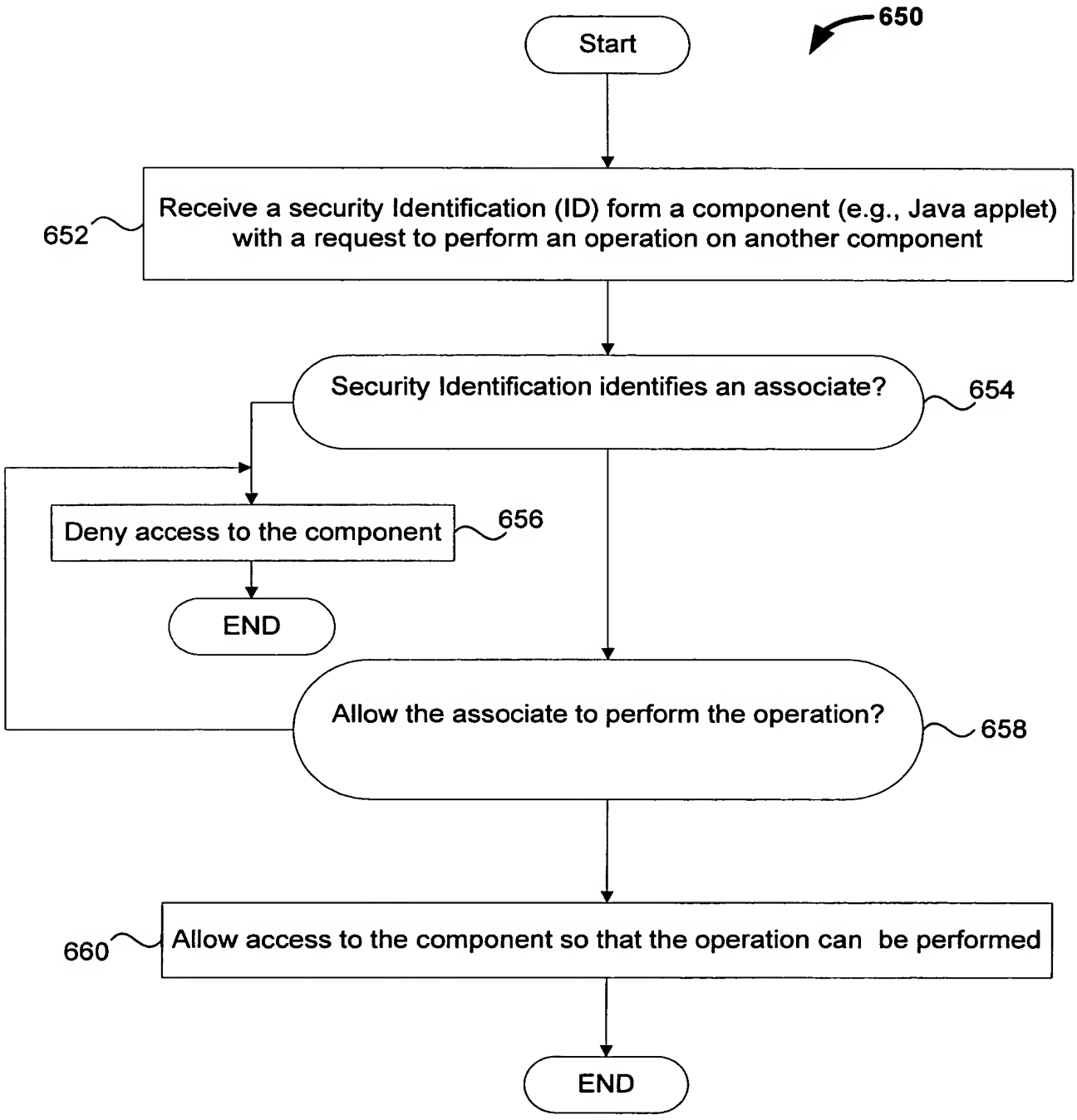

650
Start
Receive a security Identification (ID) form a component (e.g., Java applet) with a request to perform an operation on another component
652
Security Identification identifies an associate?
654
Deny access to the component
656
END
Allow the associate to perform the operation?
658
Allow access to the component so that the operation can be performed
660
END

Fig. 6F

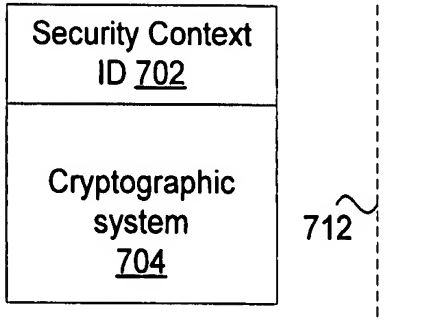


Fig. 7A

ID 1	ID 2	ID 3	ID 4	ID 5
Key 1	Key 2	Key 3	Key 4	Key 4
Key management Info 1	Key management Info 2	Key management Info 3	Key management Info 4	Key management Info 5
Algorithm(s) 1	Algorithm(s) 2	Algorithm(s) 3	Algorithm(s) 4	Algorithm(s) 4
Sign	Verify	Encrypt	Decrypt	Authentic -ation

710

712

Fig. 7B

```
class SecurityContext{
        ContextID;
        securityEnvironment[ ];
        data[ ];
        operations;
        securityContext[] (Nested?)
        }


SecurityEnvironment{
        KeyID[ ];
        Key[ ];
        keyMangementInfo[ ];
        Algorithm[ ];
}

Key {
        ID writeKey;
        ID readKey;
        ID deleteKey;
        byte [ ] keyMaterial;
}

Operations {
        sign,
        verify;
        encrypt;
        decrypt;
        authenticate;
        }
```

Fig. 7C

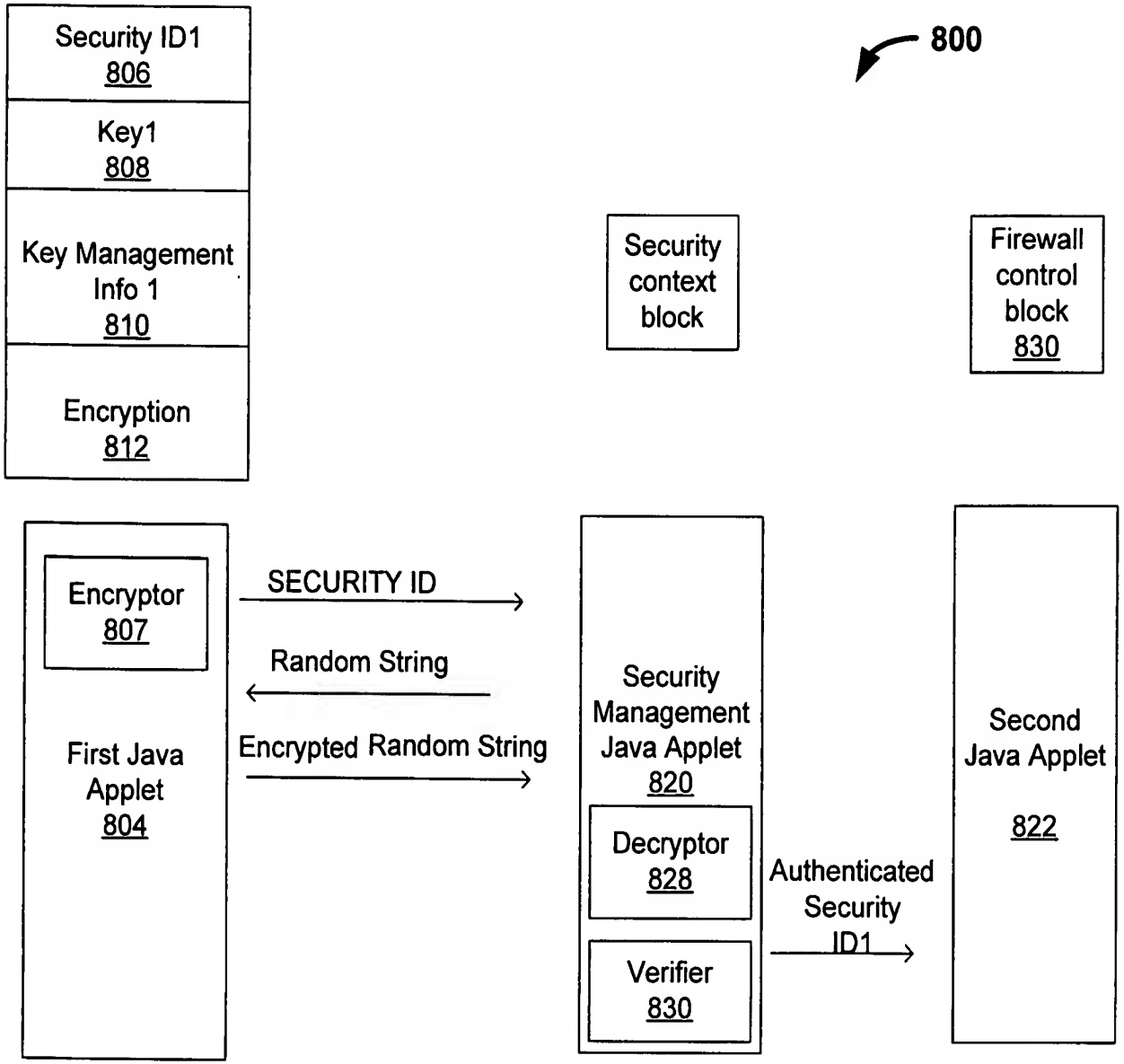

800
Security ID1
806
Key1
808
Key Management Info 1
810
Encryption
812
Security context block
Firewall control block
830
Encryptor
807
First Java Applet
804
SECURITY ID
Random String
Encrypted Random String
Security Management Java Applet
820
Decryptor
828
Verifier
830
Authenticated Security ID1
Second Java Applet
822

Fig. 8

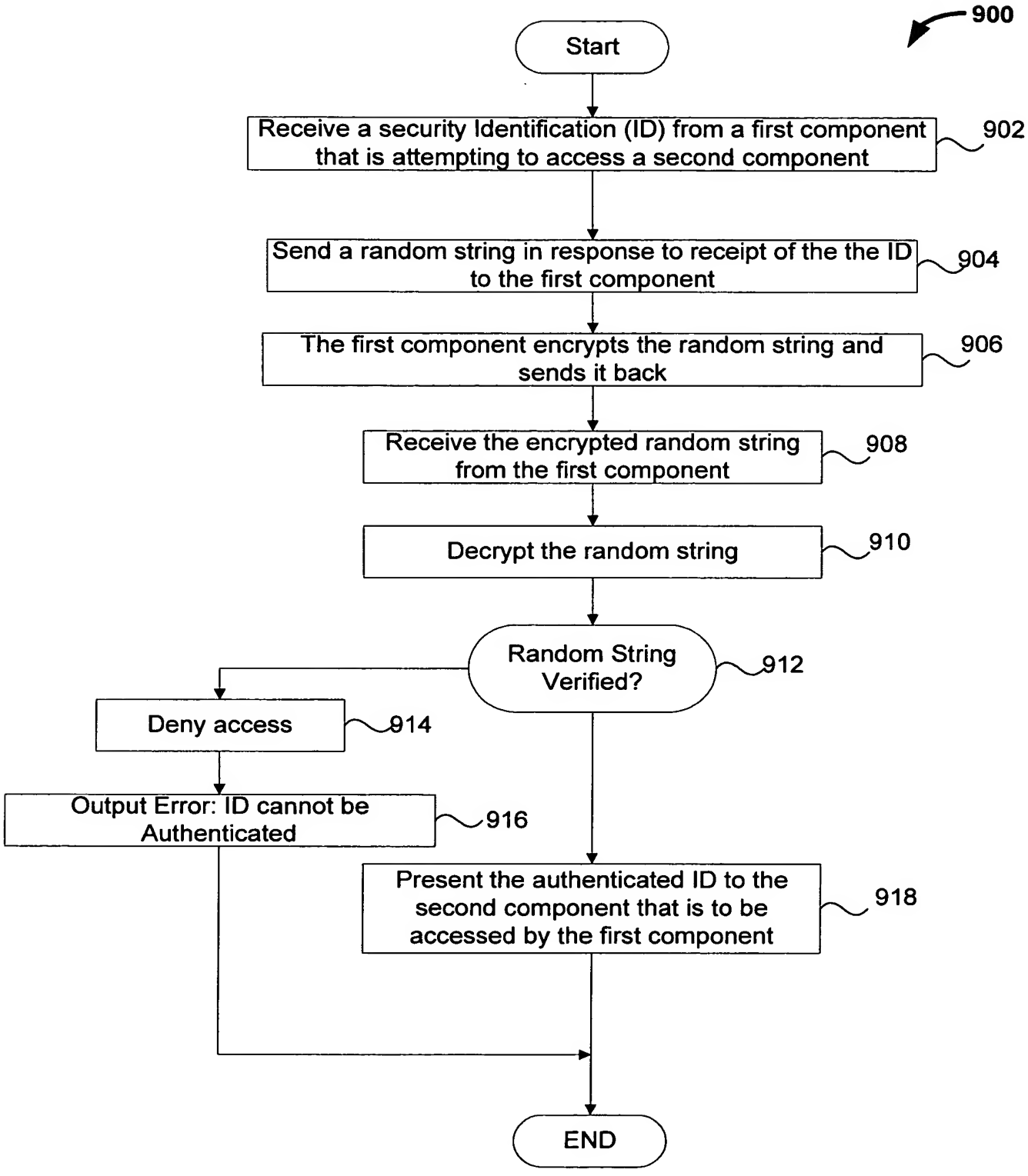

900
Start
Receive a security Identification (ID) from a first component that is attempting to access a second component
902
Send a random string in response to receipt of the the ID to the first component
904
The first component encrypts the random string and sends it back
906
Receive the encrypted random string from the first component
908
Decrypt the random string
910
Random String Verified?
912
Deny access
914
Output Error: ID cannot be Authenticated
916
Present the authenticated ID to the second component that is to be accessed by the first component
918
END

Fig. 9